

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-232247  
(43)Date of publication of application : 27.08.1999

(51)Int.Cl.

G06F 15/82  
G06F 15/82

(21)Application number : 10-028818  
(22)Date of filing : 10.02.1998

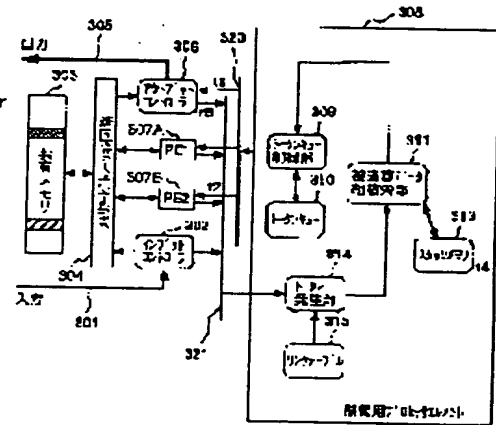
(71)Applicant : HITACHI LTD  
(72)Inventor : NAKAMOTO TAKASHI  
YAMADA HIROSHI  
NAKASE JUNKO

## (54) DATA FLOW COMPUTER AND DATA FLOW CONTROLLING METHOD

(57)Abstract:

**PROBLEM TO BE SOLVED:** To process much data at a high speed like a data flow.

**SOLUTION:** A controlling processor element 308 supplies an instruction token including flow ID indicating the contents of processing, a source pointer indicating storing position for storing the data to be processed in a shared memory 303 and a destination pointer indicating a storing position for storing the result data of processing in the shared memory 303 to an operation processor element 307A or 307B. The processor element 307A or 307B reads out data pointed by the source pointer from the memory 303, applies the processing specified by the flow ID to the read data, stores the processing result data in the storing position specified by the destination pointer in the memory 303, and transfers the instruction token to the element 308 to inform that the instruction token is processed.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(11) 特許出願公開番号

特開平11-232247

(43)公開日 平成11年(1999)8月27日

(51) Int.Cl.:

**識別記号**

**F I**

**G O 6 F 15/82**

630

**G O 6 F 15/82**

630A

640

640A

審査請求 未請求 請求項の数7 OL (全 16 頁)

(21)出願番号 特願平10-28818

(22)出願日 平成10年(1998)2月10日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 中本 貴士

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体事業部内

(72)発明者 山田 博

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体事業部内

(72)発明者 中瀬 純子

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体事業部内

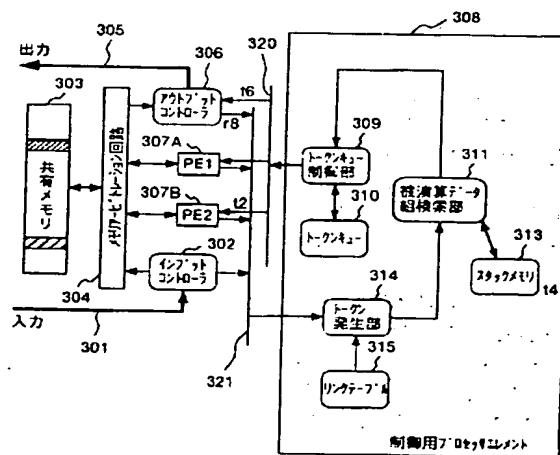
(74) 代理人 弁理士 高橋 明夫 (外1名)

(54) 【発明の名称】 データフロー計算機およびデータフロー制御方法

(57) 【要約】

【課題】多量のデータを高速にデータフロー的に処理する。

【解決手段】制御用プロセッサエレメント３０８は、処理内容を示すフローＩＤと、処理されるべきデータが記憶されている、共有メモリ３０３内の記憶位置を指すソースポイントと、処理の結果データを格納すべき、共有メモリ３０３内の記憶位置を指すデスティネーションポイントとを含む命令トークンを演算用プロセッサエレメント３０７Ａまたは３０７Ｂに供給する。そのプロセッサエレメントは、ソースポイントで指摘されたデータを共有メモリより読み出し、フローＩＤで指定される処理をそのデータに対して実行し、共有メモリ３０３内のデスティネーションポイントにより指定される記憶位置に処理結果データを格納し、上記命令トークンを制御用プロセッサエレメント３０８に転送し、その命令トークンを処理したことを通知する。



## 【特許請求の範囲】

【請求項1】複数の処理エレメントと、

これらに対する共有メモリと、

上記複数の処理エレメントに対して処理の実行を要求する命令トークンを供給する制御回路とを有し、

上記制御回路は、いずれかの処理を起動するときに、その処理に使用可能なデータが記憶された、上記共有メモリ内の記憶位置を指定する第1の情報と、上記処理の実行により得られる結果データを格納すべき、上記メモリ内の記憶位置を指定する第2の情報とを有する命令トークンを、上記複数の処理エレメントの内、その処理を実行可能な処理エレメントに供給し、

各処理エレメントは、上記制御回路により命令トークンを供給されたときに、その供給された命令トークン内の上記第1の情報により指定される、上記共有メモリ内の記憶位置から処理すべきデータを読み出し、そのデータに対して処理を実行し、その実行の結果データを、上記供給された命令トークン内の上記第2の情報により指定される、上記共有メモリ内の記憶位置に記憶するデータフロー計算機。

【請求項2】上記複数の処理エレメントは、それぞれ複数の処理を実行できるプログラム制御のプロセッサからなる請求項1記載のデータフロー計算機。

【請求項3】上記複数の処理エレメントは、それぞれ互いに異なる処理を実行する専用の回路からなる請求項1記載のデータフロー計算機。

【請求項4】上記制御回路は、プログラム制御のプロセッサからなる請求項1記載のデータフロー計算機。

【請求項5】複数の処理エレメントと、

これらに対する共有メモリと、

上記複数の処理エレメントをデータフロー制御により起動する制御回路とを有し、

上記制御回路は、いずれかの処理を起動するときに、その処理に使用可能なデータが記憶された、上記共有メモリ内の記憶位置を指定する第1の情報と、上記処理の実行により得られる結果データを格納すべき、上記メモリ内の記憶位置を指定する第2の情報とを、上記複数の処理エレメントの内、その処理を実行可能な処理エレメントに供給し、

各処理エレメントは、上記制御回路により起動されたときに、上記制御回路から供給された上記第1の情報により指定される、上記共有メモリ内の記憶位置から処理すべきデータを読み出し、そのデータに対して処理を実行し、上記制御回路から供給された上記第2の情報により指定される、上記共有メモリ内の記憶位置に、その実行の結果データを記憶するデータフロー計算機。

【請求項6】上記制御回路は、起動すべき処理エレメントに、上記第1、第2の情報と、実行されるべき処理を指定する第3の情報とを供給し、

各処理エレメントは、上記制御回路により起動されたと

きに、上記制御回路から供給された上記第3の情報により指定される処理を実行する請求項5記載のデータフロー計算機。

【請求項7】複数の処理の内、いずれか一つの処理を実行する毎に、その実行の結果生成されたデータを使用する他の処理を実行するように、複数の処理を起動するデータフロー計算機において、

いずれか一つの処理を起動するときに、その処理が使用可能なデータが記憶されている、所定のメモリ内の記憶位置を指定する第1の情報と、その処理の結果生成されるデータを記憶すべき、上記メモリ内の記憶位置を指定する第2の情報とを上記起動すべき処理に通知し、上記起動すべき処理の実行時に、上記第1の情報で指定されるデータを上記メモリから読み出し、

読み出されたデータに上記一つの処理を実行し、

その実行の結果得られたデータを、上記第2の情報により指定される、上記メモリ内の記憶位置に記憶するデータフロー制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数の処理エレメントをデータフロー制御にしたがって並列に動作させるデータフロー計算機およびデータフロー制御方法に関する。

【0002】

【従来の技術】動画像データに代表される大量のデータに対する演算を高速に処理するには、複数の処理エレメント例えばプロセッサエレメントあるいは専用の演算器を並列に動作させることが望ましい。これに適したプロセッサとしてデータフロー計算機が知られている。例えば、中野 肇監修「入門と実習 ニューロコンピュータ」(平1-9-15)技術評論社 P. 138-141参照。この種のプロセッサでは、トークンと呼ばれるデータパケットが演算の実行を制御する。トークンは、被演算データとそのデータに対する処理を指定する情報とを含む一つのまとまったデータパケットである。異なるプロセッサエレメントの間でトークンが交換される。各プロセッサエレメントは他からトークンを受信したときに、そのことを契機として、そのトークン内のデータを処理する命令あるいは命令列を実行する。その実行により生成されたデータとそのデータに対して施すべき処理を指定するための情報とを含むトークンを生成し、他のプロセッサエレメントに転送する。当該他のプロセッサエレメントも同様に受信したトークンを処理する。各プロセッサエレメントが実行すべき処理のフローを事前に定めることにより、これらのプロセッサエレメントを効率よく並列に動作させることができる。

【0003】

【発明が解決しようとする課題】従来のデータフロー計算機では、トークンには被演算データが含まれる。一つ

のトークンに長いデータを含ませた場合、トークンを、トークン制御用のプロセッサエレメントと処理エレメントの間あるいは異なる処理エレメントの間で転送するのに時間が掛かる。トークン制御用のプロセッサエレメント内での処理にも時間が掛かる。しかも、一つのトークンに含まれるデータの長さには上限があるのが普通である。したがって、処理すべきデータの全量が多くなるほど、処理すべきトークンの数が多くなる。トークンの数が増大すると、それらのトークンをトークン制御用のプロセッサエレメントと処理エレメントの間あるいは異なる処理エレメントの間で転送するのに時間が掛かる。このように、動画データに代表される多量のデータを従来のデータフロー計算機で処理させると、処理時間が長くなる。

【0004】従って、本発明の目的は、多量のデータを高速に処理可能なデータフロー計算機を提供することである。

【0005】本発明のより具体的な目的は、動画データに代表される多量のデータを高速に処理可能なデータフロー制御の信号処理プロセッサを提供することである。

【0006】

【課題を解決するための手段】本発明によるデータフロー計算機は、複数の処理エレメントと、これらに対する共有メモリと、上記複数の処理エレメントに対して処理の実行を要求する命令トークンを供給する制御回路とを有し、上記制御回路は、いずれかの処理を起動するときに、その処理に使用可能なデータが記憶された、上記共有メモリ内の記憶位置を指定する第1の情報と、上記処理の実行により得られる結果データを格納すべき、上記メモリ内の記憶位置を指定する第2の情報とを有する命令トークンを、上記複数の処理エレメントの内、その処理を実行可能な処理エレメントに供給し、各処理エレメントは、上記制御回路により命令トークンを供給されたときに、その供給された命令トークン内の上記第1の情報により指定される、上記共有メモリ内の記憶位置から処理すべきデータを読み出し、そのデータに対して処理を実行し、その実行の結果データを、上記供給された命令トークン内の上記第2の情報により指定される、上記共有メモリ内の記憶位置に記憶する。

【0007】本データフロー計算機では、処理すべきデータは命令トークンには含まれず、そのデータの記憶位置を指定する情報が含まれる。したがって、処理エレメントの間では処理の結果データは転送されない。このデータは共有メモリを介してそれらの処理プロセッサエレメント間で受け渡しされる。以上のことから、本データフロー計算機は多量のデータを高速に処理できる。

【0008】

【発明の実施の形態】<従来の技術とその問題点>本発明の実施の形態を説明する前に、前述の文献に記載の従

来のデータフロー制御のプロセッサシステムを説明する。図18に示すように、このプロセッサでは、4つのイメージパイプラインプロセッサ(ImPP)チップ(1150)の直列接続がデータ転送を司るマジックと称するチップ(1160)に接続されている。このマジックチップ1160にはイメージメモリ1170、データバッファ1180、I/Oポートデコーダ1181、DMAコントローラ1182が接続されている。各ImPPチップ1150は、図19に示すように構成されている。インプットコントローラ1104とアウトプットコントローラ1101とは、他のImPPチップおよびマジックチップ1160との間でトークンを交換する。この交換されるトークンは、データとリンク情報(ID)とを含む。インプットコントローラ1104が受信したトークンは、リンクテーブル1105、ファンクショナルテーブル1106、データメモリ1103、キュー1102、プロセッシングユニット1107により処理される。プロセッシングユニット1107による処理の結果、新たなトークンが生成され、リンクテーブル1105に供給される。リンクテーブル1105からプロセッシングユニット1107がトークンを処理するためのパイプライン・リンクを構成する。

【0009】リンクテーブル1105は、インプットコントローラ1104あるいはプロセッシングユニット1107より与えられたトークン内のリンク情報IDにより、オペレーションコードアドレス(OPA)とトークンが上記パイプライン・リンクにより処理された結果生成される次のトークンに付すべき次のリンク情報ID'との対をそのテーブル内部から読み出し、これらの新たな情報と入力トークンに含まれたデータとよりなる新たなトークンをファンクショナルテーブル1106に供給する。ファンクショナルテーブル1106は、オペレーションコードアドレス(OPA)により、そのトークンがリングを一周する間に実行すべき命令の演算コード(OP)をそのテーブルの内部より読み出す。受信したトークンが、データメモリのアドレスの生成(AG)のコードあるいはフロー制御(FC)のコードを持っている場合には、ファンクショナルテーブル1106に付属するAG-FC部(図示せず)によりそのコードが処理される。データメモリ1103は、AG-FC部により生成されたデータメモリアドレスにトークン内のデータを一時的に書き込むか、あるいはすでにこのメモリに書き込まれた常数あるいは先に書き込まれたデータをデータBとして読み出し、元のデータ(これはデータAとして扱われる)とこのデータB、リンク情報ID'、OPAを含む新たなトークンをキュー1102に供給する。キュー1102内のトークンの内、さらに処理をされるべきトークンは、プロセッシングユニット1107に供給される。このプロセッシングユニット1107はそのトークンを処理して新たなトークンを生成し、リンクテ

ープル1105に供給する。その後、以上の処理がこの供給されたトークンに対して繰り返される。その結果、キュー1102に、さらに処理を要しないトークンが格納される。このようなトークンは、アウトプットコントローラ1101によりこのプロセッサエレメントの外部に送出される。

【0010】以上から明らかなように、入力トークンに含まれたデータは、そのデータに対する処理が完了するまで、パイプライン・リング上を転送される。したがって、画像データのごとく処理すべきデータの総量が多い場合には、より多数のトークンを処理しなければならない、処理時間が掛かる。

【0011】また、複数のImPPチップ1150がシリアルに接続されているために、異なる処理を並列に実行するには、処理速度に制限があるという問題もある。

【0012】以下、本発明に係るデータフロー制御のプロセッサシステムを図面に示したいくつかの実施の形態を参照してさらに詳細に説明する。なお、以下においては、同じ参照番号は同じものもしくは類似のものを表すものとする。また、発明の第2の実施の形態以降においては、発明の第1の実施の形態との相違点を主に説明するに止める。

【0013】＜発明の実施の形態1＞図1において、データフロー制御の信号処理プロセッサは、信号処理に必要な複数の処理を並列に実行するための処理エレメントとして、いずれも複数の処理を実行可能なプログラム制御の演算用プロセッサエレメント(PE1, PE2)307A, 307Bを使用する。このプロセッサは、これらの演算用プロセッサエレメントにより共有される共有メモリ303と、このメモリへのアクセス要求のアービトレーションを行うメモリアービトレーション回路304と、インプットコントローラ302と、アウトプットコントローラ306と、制御用プロセッサエレメント308とを有する。制御用プロセッサエレメント308は、汎用のプログラム制御のマイクロコンピュータにより構成される。図においてブロック308内には、このプロセッサエレメントにより実行される、トークンの処理のための複数のプログラムモジュール314, 311, 309および制御用プロセッサエレメントのメモリ(図示せず)内に設けられたテーブル315、スタックメモリ313およびキュー310等を示している。演算用プロセッサエレメント307A, 307Bはいずれも種々の演算を実行可能な演算器(図示せず)を有する。この信号処理プロセッサは、一つのLSIチップ上に構成される。

【0014】インプットコントローラ302は、入力データを外部から受け取ると、そのデータを共有メモリ304の特定のアドレスの位置に書き込む。以下、インプットコントローラ302およびアウトプットコントローラ306がデータを外部と交換するのに用いるデータバ

ケットをデータトークンと呼び、演算用プロセッサエレメント307A, 307Bが実行するトークンを命令トークンと呼ぶ。インプットコントローラ302は、その後そのデータトークンの処理が終了したことを通知する命令トークンを生成し、制御用プロセッサエレメント308に供給する。この命令トークンには上記共有メモリアドレスが含まれる。

【0015】制御用プロセッサエレメント308は、この命令トークンに回答して、そのデータを処理するための命令トークンを発生し、演算用プロセッサエレメント307Aあるいは307Bの内、処理を実行可能な状態にある演算用プロセッサエレメントにその命令トークンを供給する。この命令トークンには、その入力データのアドレスとその入力データに対して施すべき処理を示す情報と、その処理の結果データを記憶すべき、共有メモリ303内のアドレスを含む。その演算用プロセッサエレメントは、この命令トークン内のアドレスによりその入力データを共有メモリ303から読み出し、この命令トークンにより指定された上記処理を実行し、その結果データを上記命令トークンが指定する、共有メモリ304内のアドレスの位置に書き込む。その後、この命令トークンの実行が終了したことを知らせるために、この命令トークンを制御用プロセッサエレメント308に供給する。制御用プロセッサエレメント308は、演算用プロセッサエレメント307Aあるいは307Bから供給された命令トークンに回答して、後続の命令トークンを生成する。この命令トークンは、上に述べた最初の命令トークンと同様に処理させる。以下、同様の処理が繰り返えされる。最終結果データはアウトプットコントローラ306により、共有メモリ303から読み出され外部に転送される。

【0016】このように、本実施の形態では、処理されるべきデータに代えて、そのデータのメモリアドレスが命令トークンに含まれ、演算用プロセッサエレメント307Aあるいは307Bと制御用プロセッサエレメント308との間では、データは直接には転送されない。演算結果データを生成した演算用プロセッサエレメントとそれに対して演算を行う演算用プロセッサエレメントの間では、その演算結果データはトークンを介してではなく共有メモリ303を介して受け渡される。

【0017】各演算用プロセッサエレメントが実行可能な複数の処理の例は、動画像符号化データの復号処理を行うのに必要な、可変長復号(VLD)処理、逆量子化(IQ)処理、逆DCT(IDCT)処理、動き保証(MC)である。しかし、以下の説明においては、図3に示したように、本実施の形態の信号処理プロセッサで実行される処理は、処理1から4を含む処理であると仮定し、これらの処理を引用しながら本実施の形態の信号処理プロセッサの動作を説明する。ここで、処理1は入力データAを処理し、処理3は入力データBを処理す

る。処理2は、処理1の結果データを処理し、処理4は処理2の結果データと処理3の結果データとを処理し、結果データCを生成する。処理4は、複数のデータに対する処理である。このような処理を以下では複数項演算と呼ぶことがある。例えば、算術四則演算、あるいは論理積演算あるいは論理和演算等の論理演算が複数項演算である。本実施の形態では、複数項演算が使用するデータの数は2と仮定する。

【0018】本実施の形態では、共有メモリ303のアドレスを含む命令トークンを使用するために、共有メモリ303上に各処理に対応して、その処理が使用するデータを格納する領域を予め決めておく。例えば、図4に示すように、領域aは処理1で使用される領域であり、入力データAとBは、それぞれこの領域a内のアドレスa1とa2の領域に記憶される。領域bは処理2により使用される領域であり、処理1の結果データがこの領域b内のアドレスb1の領域に記憶される。領域cは処理4により使用される領域であり、処理2と3の結果データがそれぞれこの領域c内のアドレスc1とc2の領域に記憶される。領域dは、アウトプットコントローラ30により使用される領域であり、具体的には処理4の結果データCがこの領域d内のアドレスd1の領域に記憶される。これらの領域aからdは、互いに他と重複しないように定められる。

【0019】本実施の形態で使用される命令トークンは、図2のフォーマット321に示されるように、フロー識別子（フローID）、演算タグ、ソースポインタ、デスティネーションポインタという4つのフィールドからなる。フローIDは処理の種類を表す。演算タグは、フローIDが指定する処理が複数のデータを使用する処理（複数項演算）であるときに、それらのデータを区別するためのものである。例えば、同じ処理に使用される最初のデータに対しては番号1を使用し、同じ処理に使用する第2のデータに対しては番号2を使用する。一つの処理が使用するデータが一つの時には、そのデータに対しては演算タグは使用されない。ソースポインタは、処理されるべきデータが記憶されている、共有メモリ303上のアドレスを示す。デスティネーションポインタは、その命令トークンが要求する処理の結果データを格納すべき、共有メモリ303上のアドレスを示す。

【0020】本実施の形態で使用されるデータトークンは、図2のフォーマット322で示されるように、データの入力を示すフローIDと、処理されるべきデータとを含む。

【0021】インプットコントローラ302は、線301を介してデータトークンを外部から受信し、このトークン内のデータを共有メモリ303に書き込むとともに、そのデータトークンの処理が終了したことを通知するための命令トークンを生成してトークン発生部314に転送する回路である。図3の処理1が使用すべきデー

タAが入力されデータトークンの場合、インプットコントローラ302は、共有メモリ303内の領域aに属する領域a1（図4）のアドレスを作成し、そのデータトークンを受信したときに、このアドレスを使用してデータトークンに含まれたデータAをこの領域a1に記憶する。同様に、処理3が使用すべきデータBが入力されるデータトークンの場合、インプットコントローラ302は、そのデータトークンを受信したときに、領域a2のアドレスを作成し、データBをこの領域a2に記憶する。なお、インプットコントローラ302は、共有メモリ303に対するアクセスを監視し、そこからデータが読み出されたときにはその読み出し位置を検出し、この検出結果に基づいて、入力されたデータAあるいはBに対する処理が終了したことを判断し、領域a内の空き領域を検出する回路を有する。インプットコントローラ302はこの検出された空き領域に入力データを書き込む。

【0022】入力データの書き込みの完了後に、インプットコントローラ302は書き込みの完了を通知するための命令トークンを生成し、トークン発生部314に供給する。この命令トークンでは、フローIDは、データAまたはBの入力を示し、デスティネーションポインタは、その入力データが書き込まれた、共有メモリ303内のアドレスを示す。ソースポインタおよび演算タグは有効でない。

【0023】図5にインプットコントローラ302の概略構造を示す。入力データはFIFO（1801）に一旦格納される。書き込みを開始するアドレスを書き込み開始アドレスレジスタ1805と書き込みアドレスレジスタ1804に格納し、共有メモリ803に書き込みアドレスとして供給する。前記FIFO（1801）に格納されたデータは、レジスタ1802を経由して共有メモリ803（図13）に供給され、レジスタ1804内のアドレスを用いて、共有メモリ803が許す期間に格納される。インクリメント1803により書き込みアドレスレジスタ1804内のアドレスを逐次インクリメントする。以上の動作は入力データの後続の部分に対して逐次繰り返される。こうして、多量の入力データが逐次共有メモリ803に書き込まれる。比較器1807は、レジスタ1805内の書き込み開始アドレスとレジスタ1804内の現在の書き込みアドレスとを比較することにより、所定量のデータの書き込みが終了したか否かを示す比較結果を出力する。制御回路1808はこの出力が所定量のデータの書き込みの完了を示すときに、書き込み動作を終了し、レジスタ1806内に入力完了を示すフローIDを有する命令トークンを生成し、バス820を介して制御用プロセッサエレメント308に供給する。このトークン内のデスティネーションにはレジスタ1805内の書き込み開始アドレスが使用される。なお、インプットコントローラ302に含まれた空き領域

を検出する前述の回路は簡単化のために図示していない。

【0024】トークン発生部314は、インプットコントローラ302あるいは演算用プロセッサエレメント307Aあるいは307Bから供給される命令トークンにตอบสนองして、リンクテーブル315を参照しながら新たな命令トークンを発生するプログラムモジュールである。この新たな命令トークンは、供給された命令トークンが通知する、実行が終了した処理の次に実行すべき処理を指定する命令トークンである。トークン発生部314は、生成した命令トークンを被演算データ組検索部311に転送する。

【0025】リンクテーブル315は、トークン発生部314に供給される可能性があるいろいろの命令トークンが指定する処理に対応して一つのエントリを有し、そのエントリには、その処理の次に実行されるべき処理に割り当てられたフローIDと、当該次の処理を受けるべきデータに付すべき演算タグを予め記憶する。図7には、図3に例示された処理を実行するときにリンクテーブル315に記憶される内容を示す。例えば、トークン発生部314に供給された命令トークンが、データAあるいはBの入力を示すとき、次の処理は、それぞれ処理1または3である。後に説明するようにトークン発生部314に処理1の実行の完了を示す命令トークンが供給されたとき、次に実行すべき処理は処理2である。次に実行すべき処理が処理1、2または3であるとき、これらの処理はいずれも一つのデータのみを使用するので、これらの処理を指定する命令トークンには演算タグは使用されない。トークン発生部314に処理2または3の実行の完了を示す命令トークンが供給されたとき、次に実行すべき処理は処理4である。この処理は二つのデータを使用する複数項演算である。したがって、これらの処理2と3のいずれか一方の実行完了を示す命令トークンがトークン発生部314に供給されたとき、次に実行すべき処理4を示す命令トークンに含まれるべき演算タグは1となる。処理2と3の他方の実行完了を示す命令トークンがトークン発生部314に供給されたとき、次に実行すべき処理4を示す命令トークンに含まれるべき演算タグは2となる。最後に、処理4の実行が完了したときに、次に実行すべき処理は結果データCの外部への出力である。このデータ出力を示す命令トークンにも演算タグは使用されない。トークン発生部314は、この命令トークンを、処理4の実行完了を示す一組の命令トークンがトークン発生部314に供給されたとき生成するようになっている。

【0026】具体的には、トークン発生部314は、リンクテーブル315を参照して図6に示すように動作する。すなわち、通常は命令トークンの到着待ちの状態にあり、命令トークンが到着したか否かを繰り返しチェックしている。命令トークンが到着すると、リンクテー

ブル315内のその命令トークンが指定する処理に対応するエントリから、その処理の次の処理に対するフローIDを読み出す。このフローIDが複数項演算に対するものであるか否かを判別し、もしそうであればそのエントリからタグを読み出す。発生した命令トークンに含まれたデスティネーションポインタを上記次の処理のためのソースポインタとして取り出す。さらに、後に説明する方法により上記次の処理のためのデスティネーションポインタを決定する。取り出されたフローID、演算タグ、ソースポインタおよび決定されたデスティネーションポインタを用いて、上記次の処理のための命令トークンを発生し、被演算データ組検索部311に転送する。その後、トークン発生部314は待機状態に戻る。

【0027】トークン発生部314は、図8に示す手順にて上記次の処理のためのデスティネーションポインタを決定する。トークン発生部314は、共有メモリ303中の使用中の領域のアドレスを保持する図示しないアドレススタックを有する。したがって、このアドレススタックには、トークン発生部314に到着した命令トークン内のソースポインタがすでに格納されている。この命令トークンの到着時点ではこのソースポインタを有する記憶領域はすでに使用済みとなっている。したがって、このソースポインタを上記アドレススタックから削除する。上記命令トークン内のデスティネーションポインタを、上記次の処理のためのソースポインタとして取り出した後に、この新たなソースポインタをこのアドレススタックに格納する。その後、上記次の処理の結果データを格納する領域として共有メモリ303内の空き領域、すなわち、このアドレススタックに登録されたアドレス以外のアドレスを有する領域を検索する。検索された領域のアドレスを上記次の処理のためのデスティネーションポインタとして使用する。さらに、このデスティネーションポインタを上記アドレススタックに登録する。

【0028】被演算データ組検索部311は転送された命令トークンが複数項演算を要求するとき、その命令トークンが必要とする二つのデータがそろうまで、その命令トークンをスタックメモリ313に格納し、後に同じ複数項演算を要求する命令トークンがトークン発生部314から供給されたときに、これらの二つの命令トークンを実行可能な命令トークンとしてトークンキュー制御部309に転送する。もしトークン発生部314から供給された命令トークンが、複数項演算を要求しないときには、被演算データ組検索部311は、その命令トークンをそのまま実行可能な命令トークンとしてトークンキュー制御部309に転送する。

【0029】より具体的には、被演算データ組検索部311は図10に示すように動作する。まず被演算データ組検索部311は命令トークンの到達待ち状態にあり、トークン発生部314から命令トークンが到着したか否



かを検出している。命令トークンが到着すると、トークン内のフローIDにより到着した命令トークンが複数項演算を要求するか否かを判別する。もし、命令トークンが複数項演算を要求しないときには、到着した命令トークンをそのままトークンキュー制御部309に出力する。命令トークンが複数項演算を要求するときには、到着した命令トークンと組となるべき他の命令トークンをスタックメモリ313から検索する。この他の命令トークンは、到着した命令トークンと同じフローIDを有し、異なる演算タグを有するものである。組となるべき他の命令トークンがスタックメモリ313内に存在しなければ、転送されてきた命令トークンをスタックメモリ313に格納する。組となるべき他の命令トークンがスタックメモリ313内に存在したときには、当該他の命令トークンをスタックメモリ313から読み出し、到着した命令トークンとともに実行可能な命令トークンとしてトークンキュー制御部309に転送する。

【0030】トークンキュー制御部309は、図10に示すように、命令トークンの待ち合わせルーティン(図10(a))と命令トークンの発行ルーティン(図10(b))とを実行する。命令トークンの待ち合わせルーティンでは、トークンキュー制御部309は、命令トークンが被演算データ組検索部311から到達することにより、到着した命令トークンをトークンキュー310に格納する。命令トークンの発行ルーティンでは、トークンキュー制御部309は、演算用プロセッサエレメント307A、307Bの動作状況を監視し、処理の実行を終了し待機状態に入った演算用プロセッサエレメントがあれば、トークンキュー310から演算用プロセッサエレメントにより処理されるべき命令トークンを取り出し、その演算用プロセッサエレメントにバス320を介して転送する。なお、トークンキュー310に保持された命令トークンが、複数項演算を要求するときには、その命令トークンと、その命令トークンが要求する処理と同じ処理を要求し、演算タグの値が異なる他の命令トークンとの組を同じ演算用プロセッサエレメントに供給する。なお、全ての演算用プロセッサエレメント307A、307Bがいずれもビジー状態にある間は、トークンキュー310に格納された命令トークンはそのまま保持される。

【0031】演算用プロセッサエレメント307Aまたは307Bは複数の命令トークンにより指定される複数の処理をすべてを実行できるものである。トークンキュー制御部309から命令トークンが供給されると、演算用プロセッサエレメント307Aまたは307Bは、その命令トークン内のソースポイントにより指定されるデータを共有メモリ303から読み出す。その命令トークンが複数項演算を要求するときには、その命令トークンに含まれた第1のソースポイントと、その命令トークンと対をなす他の命令トークンに含まれた第2のソース

ポイントにより指定される二つのデータを共有メモリ303より読み出す。読み出された一つまたは二つのデータに対してその命令トークンが要求する処理を実行し、共有メモリ303内の、その命令トークンのデスティネーションポイントにより指定された記憶領域に、実行の結果得られたデータを格納する。その処理の実行完了後に、上記処理された命令トークンをトークン発生部314にバス321を介して供給する。トークン発生部314は、既に述べたと同様にしてこの命令トークンの後続の命令トークンを発生する。こうして、インプットコントローラ302に外部より供給されたデータを処理するために一連の命令トークンが生成され、実行される。その結果、最終的には外部にデータを出力することを要求する命令トークンがトークン発生部314により生成され、トークンキュー制御部309に転送されることになる。

【0032】トークンキュー制御部309は、トークンキュー310内にデータ出力を指示する命令トークンが含まれているときには、その命令トークンをアウトプットコントローラ306に供給する。アウトプットコントローラ306は、供給された命令トークン内のソースポイントで指定されるデータを共有メモリ304から読み出し、線305を介して外部に転送する。

【0033】図11は、このアウトプットコントローラ306の概略構成を示す。デコーダ1902が到達したトークン1901のフローIDを解読した結果、この命令トークンが出力処理を要求することを確認する。読み出しアドレスレジスタ1904には、このトークン内のソースポイントをセレクト1908を介してセットし、読み出しアドレスとして共有メモリ803に供給し、共有メモリ803上のソースポイントが示す記憶位置からデータを読み出す。読み出されたデータはレジスタ1907を介してFIFO(1906)に格納される。FIFO(1906)に格納されたデータは出力データとして外部に転送される。その後、インクリメント1905とセレクト1908を使用してレジスタ1904内の読み出しアドレスを繰り返し更新することにより、所定数の後続のデータを共有メモリ803から読み出し、外部に供給する。

【0034】次に具体的に図3で示す処理1、処理2、処理3、処理4からなる一連の処理を本発明の信号処理プロセッサで実行した場合の動作を図12を参照して説明する。インプットコントローラ302はデータAを含むデータトークンを外部から受信すると、入力データ領域a(図4)内の現在使っていない記憶位置にデータAを格納し、さらにこのデータの入力を通ずるトークンr1を発生する。同様に、データBを含むデータトークンを受信すると、入力データ領域a内の現在使っていない場所にデータBを格納し、このデータの入力を通ずる命令トークンr2を発生する。トークン発生部314

では、インプットコントローラ302から発生された命令トークンr1、r2のそれぞれに対して次の処理と共有メモリ303上のデスティネーションポイントとを割り付け、それぞれ処理1、3の実行を要求する命令トークンt1、t2を発生する。

【0035】命令トークンt1、t2は今の例では複数項演算を要求しないので、被演算データ組検索部311はそのままこれらの命令トークンt1、t2をトークンキュー制御部309に送り、トークンキュー310に格納する。トークンキュー制御部309は、演算用プロセッサエレメント307A、307Bの動作状況を監視し、処理を実行可能な演算用プロセッサエレメントにこれらの命令トークンを転送する。今の場合、プロセッサエレメントPE1に命令トークンt1を転送し、プロセッサエレメントPE2に命令トークンt2を転送すると仮定する。プロセッサエレメントPE1、PE2は転送された命令トークンt1、t2に基づき処理1、処理3をそれぞれ共有メモリ303をアクセスしながら実行する。プロセッサエレメントPE1、PE2は、それぞれ命令トークンt1、t2の実行が終了したときに、それぞれの命令トークンをそのまま命令トークンr3、r7としてトークン発生部314に転送する。

【0036】トークン発生部314は命令トークンr3、r7を受け、処理2、4をそれぞれ要求する命令トークンt3、t4を新たに発生する。処理2のトークンt3は前記トークンt1、t2と同様にトークンキュー310を経て演算用プロセッサエレメント、例えばPE1に送られ、そこで実行される。その後、命令トークンr4がトークン発生部314に転送される。一方、処理4を行う命令トークンt4は複数項演算を要求するので、被演算データ組検索部311が組となる、処理4を要求し、もう一つの演算タグがついた命令トークンをスタックメモリ313から検索する。今の場合、このような命令トークンはスタックメモリ313にはまだ到達していないため、被演算データ組検索部311は命令トークンt4をスタックメモリ313に格納する。

【0037】一方、命令トークンr4を受けたトークン発生部314は次の処理を実行するための命令トークンとして、処理4を要求する命令トークンt5を発生する。被演算データ組検索部311は、命令トークンt5と組となるタグ2の命令トークンを検索する。ここでは、スタックメモリ313にすでに格納された命令トークンt4を発見し、命令トークンt4、t5を実行可能な命令トークンとしてトークンキュー310に転送する。トークンキュー制御部309は、処理を実行可能な演算用プロセッサエレメント、例えばPE1にこれらの一組の命令トークンを転送する。その演算用プロセッサエレメントPE1は、これらの一組の命令トークンを用いて処理4を実行したのち、その一組の命令トークンをそのまま一組の命令トークンr5、r6としてトークン

発生部314に転送する。トークン発生部314はこれらの一組の命令トークンr5、r6を受けてアウトプットコントローラ306にデータ出力を指示する命令トークンt6を発行する。アウトプットコントローラは命令トークンt6を受けデータの出力を行う。

【0038】以上に説明したように、本実施の形態では、演算用プロセッサエレメント307A、307Bは、共有メモリ303上の被演算データに対し演算を行い、共有メモリ303に演算結果データを格納するが、演算結果データを演算用プロセッサエレメント間で伝達されることがない。したがって、多量のデータを高速に処理することができる。従来のように命令トークンに処理されるべきデータを含める方法では、一つのトークンに含まれるデータの長さに制限があり、長いデータを処理する場合には命令トークンの数を増大する必要がある。しかし、本実施の形態のように、共有メモリ303を介してデータを転送する方法では、一時に転送できるデータの長さはそのような制限された長さより長くできる。したがって、多量のデータを処理するときでも、トークンの数を増やす必要がなく、しかもデータを制御用プロセッサエレメント内のいろいろのモジュールの間を転送する必要はない。これらのことより、本実施の形態では、多量のデータを高速に処理できる。また、データ転送に関わる演算用プロセッサエレメント間のアービトレーションの必要がなく、共有メモリ303のデータを管理するだけで良い。

【0039】＜発明の実施の形態2＞本実施の形態は、複数の処理エレメントとして、実施の形態1で使用された演算用プロセッサエレメントに代えて、限定された処理のみを行うことが出来る専用演算器、具体的にはアクセラレータを用いるデータフロー制御の信号処理プロセッサを提供する。図13に示すように、このプロセッサは、MPEG (Moving Pictures Experts Group) の復号処理を行うもので、専用演算器として可変長復号 (VLD) 処理のアクセラレータ (VLDアクセラレータ) 807A、逆量子化 (IQ) 処理のアクセラレータ (IQアクセラレータ) 807B、逆DCT (IDCT) 処理のアクセラレータ (IDCTアクセラレータ) 807C、動き補償 (MC) 処理のアクセラレータ (MCアクセラレータ) 807D、インプットコントローラ805、アウトプットコントローラ806を持ち、これらの回路はメモリバススイッチ804を介して共有メモリ803に接続されている。図示のプロセッサは望ましくは一つのLSIチップ上に構成される。共有メモリ803は別々のメモリ803Aから803Fにより構成され、それぞれは互いに並列にアクセス可能である。入力データ801はインプットコントローラ805により共有メモリ803に書き込まれ、また出力データはアウトプットコントローラ806により共有メモリ803から読み出される。

【0040】各アクセラレータは、制御用プロセッサエレメント308から転送される命令トークンにより指定されるデータを共有メモリ803から読み出し、各アクセラレータが担当する処理をそのデータに対して施す。その結果データを再び共有メモリ803に格納し、その命令トークンをそのまま処理が終了した命令トークンとして制御用プロセッサエレメント308に転送する。命令トークンの生成と実行タイミングの制御は制御用プロセッサエレメント308が行う。制御用プロセッサエレメント308は、実施の形態1(図1)と同様に、トークン発生部314、リンクテーブル315、被演算データ組検索部311、トークンキュー制御部309、トークンキュー310を含む。実施の形態1と異なり、トークンキュー制御部309は、実行可能な命令トークンが指定する処理を実行できる特定のアクセラレータが処理を実行可能な状態にあるか否かを判別し、そのアクセラレータが実行可能なときに、その命令トークンをそのアクセラレータに供給する。

【0041】図14に、図13の信号処理プロセッサにより実行される復号処理のフローを示す。入力データAに対して可変長復号処理(VLD)が行われる。その結果データBに対して逆量子化処理(IQ)が行われる。その結果データCに対して逆DCT(IDCT)処理が行われる。その結果データDと予め共有メモリ上に格納されたデータEに対して動き補償処理(MC)が行われる。これは以前の処理結果データを再び使用する動き補償処理の特徴による。その結果データFは最終的に信号Gとして外部に出力される。

【0042】以上の復号処理の実行のために発行される命令トークンとそれらの命令トークンが処理するデータが記憶されているメモリを図15に示す。以下、図15を参照して、図14の処理の実行の模様を説明する。まず、データAが入力されると、インプットコントローラ805は、メモリ803A上のアドレスa1の位置にデータAを格納し、データの入力の完了を通知する命令トークンt1を発生し、制御用プロセッサエレメント308に供給する。

【0043】図15に戻り、命令トークンt1を受けた制御用プロセッサエレメント308は次の処理を指定する命令トークンt2を発生し、VLDアクセラレータ807Aに転送する。命令トークンt2はVLD処理をメモリ803A上のアドレスa1のデータに対して行い、メモリ803B上のアドレスb1の位置に結果データBを格納することを要求する。VLDアクセラレータ807Aは命令トークンt2で指定される処理を行い、処理が終了した段階で命令トークンt2をそのまま命令トークンt3として制御用プロセッサエレメント308に転送する。

【0044】命令トークンt3を受けた制御用プロセッサエレメント308は次の処理を実行する命令トークン

t4を発行する。命令トークンt4はIQ処理をメモリ803B上のアドレスb1のデータに対して行い、メモリ803C上のアドレスc1の位置に結果データCを格納する命令である。IQアクセラレータ807Bはこの命令トークンt4で指定される処理を行い、処理が終了した段階で命令トークンt4をそのまま命令トークンt5として制御用プロセッサエレメント308に転送する。

【0045】命令トークンt5を受けた制御用プロセッサエレメント308は次の処理を実行する命令トークンt6を発行する。命令トークンt6はIDCT処理をメモリ803C上のアドレスc1のデータに対して実行し、メモリ803D上のアドレスd1の記憶位置に結果データDを格納する命令である。IDCTアクセラレータ807Cはこの命令トークンt6で指定される処理を行い、処理が終了した段階で命令トークンt6をそのまま命令トークンt7として制御用プロセッサエレメント308に転送する。

【0046】命令トークンt7を受けた制御用プロセッサエレメント308は次の処理を実行する命令トークンt8を発行する。命令トークンt8は2つの命令トークンからなる。MC処理は、複数項演算を要求し、これらの2つの命令トークンはMC処理に使用されるべき複数のデータの一つを指定する。すなわち、一つの命令トークンは、結果データDに対してMC処理を実行することを要求し、他の命令トークンは、メモリ803F上のアドレスe1の位置のデータEに対しMC処理を行うことを要求する。MCアクセラレータ807Dはこの命令トークンt8で指定される処理を行い、処理が終了した段階で命令トークンt8をそのまま命令トークンt9として制御用プロセッサエレメント308に転送する。図15ではメモリ803からのデータの読み出しを実行する命令トークンは簡単化のために示していない。

【0047】命令トークンt9を受けた制御用プロセッサエレメント308は最後の処理を実行する命令トークンt10を発行する。命令トークンt10はデータ出力をメモリ803F上のアドレスg1のデータに対して行う命令である。アウトプットコントローラ809はこの命令トークンt10で指定されるデータ出力を行い、処理が終了した段階で命令トークンt10をそのまま命令トークンt11として制御用プロセッサエレメント308に転送する。

【0048】以上の処理で入力されたデータに対する一連の処理が終わるが、実際にはこの一連の処理が行われている最中に次の入力データがインプットコントローラ805に入り、それに対する一連の処理をアクセラレータ807Aから807Dにより並列に実行されることになる。

【0049】上述の動作説明から明らかなように、各アクセラレータの動作タイミングは、そのアクセラレータ

が処理できる特定の処理を要求する命令トークンの到着に依存している。このことが各アクセラレータの並列動作の効率を通常のパイプライン処理に比較して高いものとし、信号処理プロセッサ全体の処理能力を上げる結果につながっている。このことを示しているのが図16のタイミング図である。ある処理を単一のアクセラレータが実行し、次の処理を別のアクセラレータが実行する場合、各アクセラレータをパイプライン動作にして動作させるのが一般的である。このようなパイプライン動作を実現させるためには、各アクセラレータが処理に要する時間の内、最長期間を見積もり、それを一サイクルとしてパイプライン動作を行わせることが多い。図16の上側に示した通常のパイプライン制御のタイミングは、最初にVLD処理(VLD1)を実行する3サイクル中に、IQ処理およびIDCT処理をそれぞれ四回行うことを基本にしてパイプライン動作を設計したときのタイミングチャートである。このような設計によれば、いずれかのVLD処理が、例えば2番目に行われたVLD処理のように、2.5サイクルで終了したとしても、次のVLD処理を行うのは2番目のVLD処理の3サイクル後に行うのが通常である。

【0050】これに対して、本実施の形態のように各アクセラレータの動作開始を命令トークンの到着で制御する場合には、各アクセラレータは、被演算データが準備出来ていれば、サイクルの制約なしに処理の実行を開始できる。すなわち、図16の下側に示したように2番目のVLD処理が2.5サイクルかかって終了した場合、第3のVLD処理をすぐに開始できる。このように本実施の形態によれば、一サイクル以内の無駄な待ち時間を削減することができ、全体の処理速度を向上させることが可能である。このことは実施の形態1にもあてはまる。逆に実施の形態1と同様に、本実施の形態でもデータを命令トークンに含めないで、共有メモリを介して異なる処理エレメント(アクセラレータ)間で転送する。したがって、より多量のデータを高速に処理可能である。

【0051】＜発明の実施の形態3＞実施の形態1と2では、トークン発生部314がいずれかの命令トークンを受信したときに、トークン発生部314はその命令トークンが指定する処理の次の処理としてリンクテーブル315に予め記憶された処理を要求する命令トークンを生成した。したがって、前述の信号処理プロセッサにより実行される処理の流れは固定的であった。しかし、このような処理の流れを動的にすることもできる。

【0052】すなわち、ある処理エレメント(実施の形態1における演算用プロセッサエレメントあるいは実施の形態2におけるアクセラレータ)は、いずれかの処理を実行したときに、そのことを知らせる命令トークンをトークン発生部314に供給していた。この命令トークンにフラグを付加し、その値を適宜変更することによ

り、処理の流れを変更できる。すなわち、そのフラグの値により二つの処理の流れの一方に分岐できるようにする。具体的には、処理1の結果データに対して処理3を行い、その結果データBを出力する処理の流れと、処理1の結果データに対して処理2を行った後に処理3を行い、結果データBを出力する処理の流れとに分岐可能である。この分岐は以下のようにして実現される。

【0053】図17にこのフラグを利用した処理フローの変更の例を示す。この例では、ある処理エレメントの変更に例を示す。この例では、ある処理エレメントは、そこに供給された命令トークン1に従って、入力データAに対して処理1を実行し、その命令トークン1にフラグを付けて新たな命令トークン2を生成し、トークン発生部314に転送する。このフラグの値は、処理の実行の結果データに依存して値1または0を採用する。トークン発生部314は、このフラグの値が0か1かにより命令トークン3または4を発行する。命令トークン3は、処理1の実行結果に対して処理3の実行を要求する命令トークンである。このように異なる処理を要求する命令トークンを切り替えて発行するには、リンクテーブル315に、同一の処理の後に実行すべき次の処理として、フラグの値により選択可能な複数の処理を記憶させておく。

【0054】もし、命令トークン2に対して命令トークン3が発生された場合、この命令トークン3を転送された適当な処理エレメントは、処理1の結果データに対して処理3を実行し、その結果、命令トークン7を出力する。この処理の流れでは、処理3の実行結果データがデータBとして使用される。一方、命令トークン2に対して命令トークン4が発生された場合、この命令トークン4が転送された適当な処理エレメントは、処理1の結果データに対して処理2を実行し、命令トークン5を発生する。トークン発生部314は、この命令トークン5を受け取ったときに処理2の実行結果データに対して処理3を実行することを要求する命令トークン6を発行する。ある処理エレメントは、命令トークン6に従って処理3を実行する。このように、第2の処理の流れでは処理2が余分に実行される。

【0055】＜変形例＞本発明は以上の実施の形態に限定されるのではなく、以下に例示する変形例および他のいろいろの変形例として実施可能である。

【0056】(1)実施の形態1では二つの演算用プロセッサエレメントを使用した。演算用プロセッサエレメントを増やすことが可能である。このとき、演算用プロセッサエレメントを制御しているトークン発生部314、トークン検索部、トークンキュー等の構成をまったく変更することは必要はない。演算用プロセッサエレメントの数を増やすことにより並列に実行できる処理が増え、高速に多量のデータの演算を行うことが可能となる。

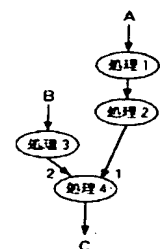
【0057】(2)実施の形態1あるいは2では、制御

【図5】図1の装置で使用されるインプットコントローラの概略構成図。

【図19】図18の装置に用いるプロセッサの概略構成図。

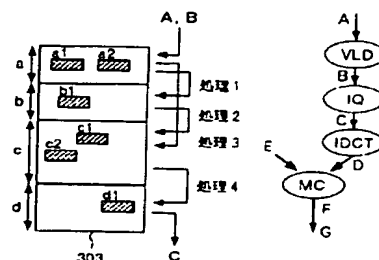
【図3】

**3**



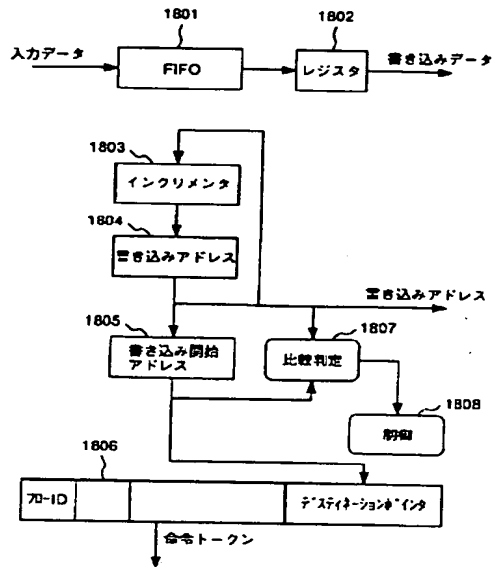
【図14】

14



【図5】

図 5



【図7】

図 7

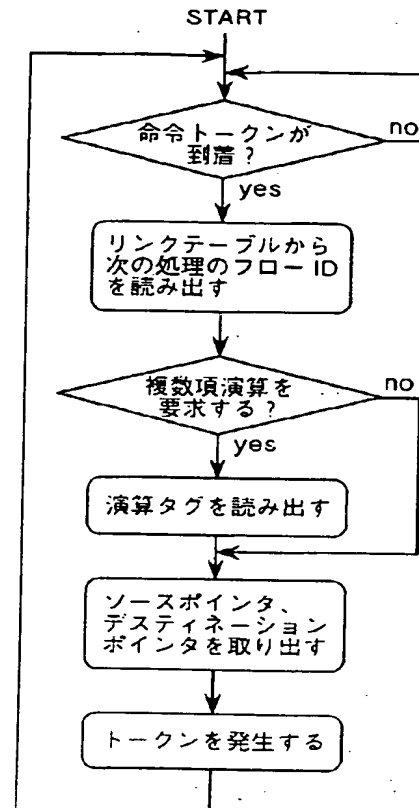
315 リンクテーブル

現在フローID	次処理のフローID	タグ
入力A	処理1	—
入力B	処理3	—
処理1	処理2	—
処理2	処理4	1
処理3	処理4	2
処理4	出力	—

【図6】

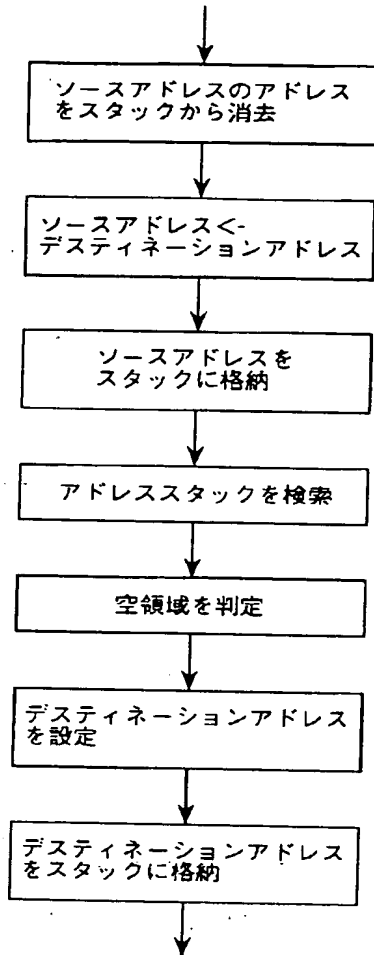
図 6

314 トークン発生部



【図8】

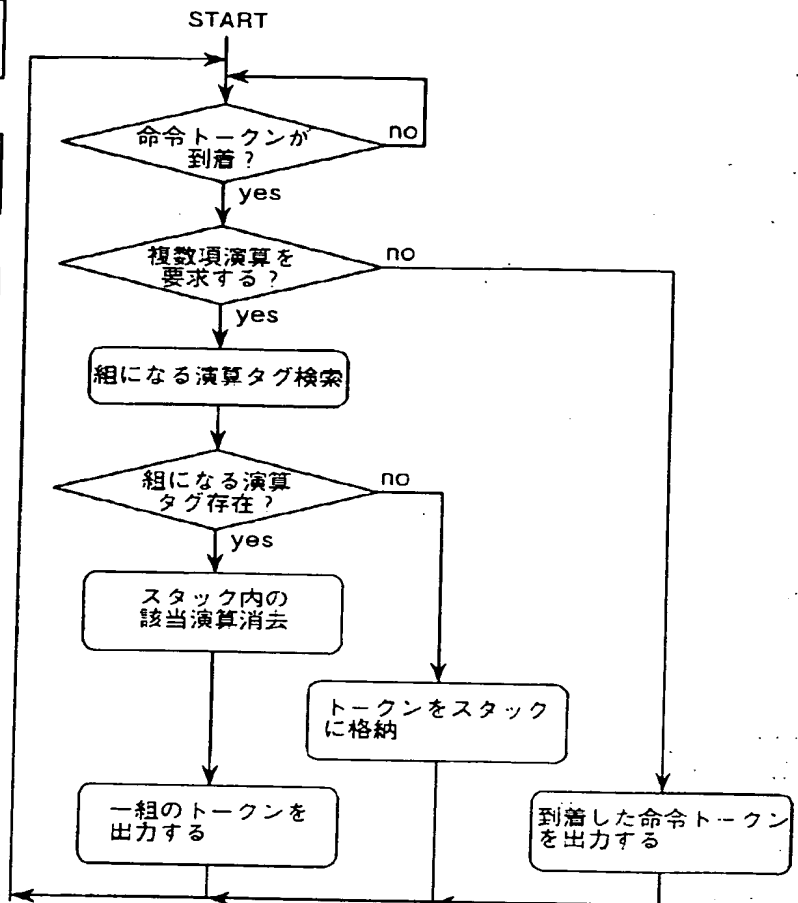
図 8



【図9】

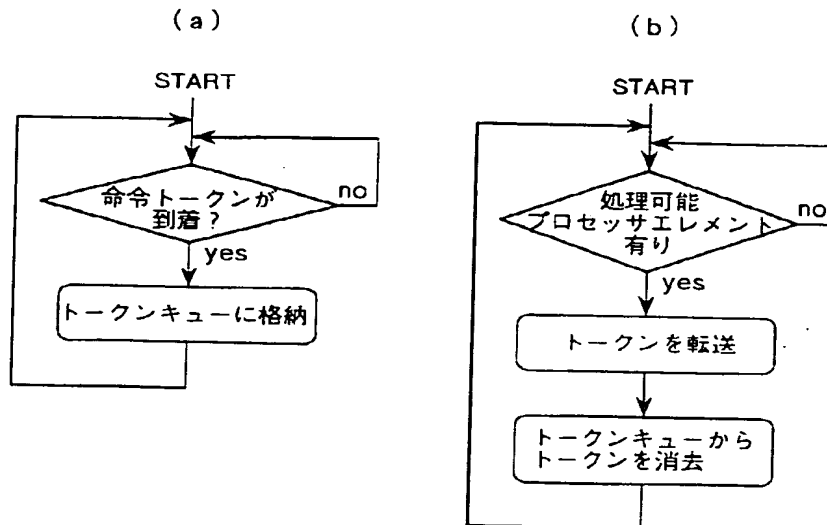
図 9

## 311 被演算データ組検索部



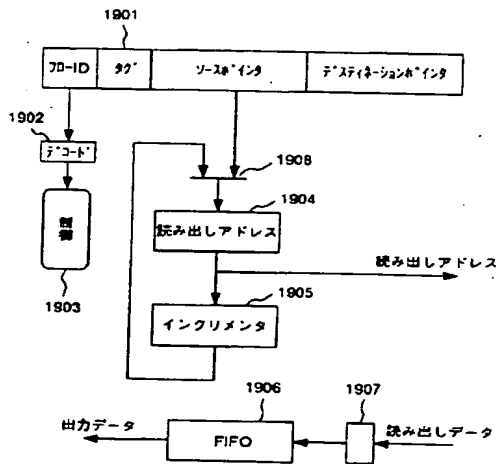
【図10】

図 10



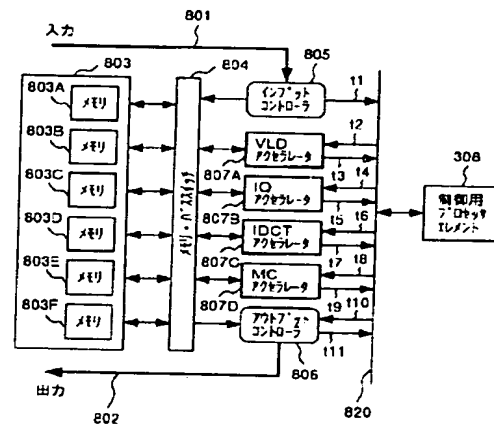
【図11】

図 11



【図13】

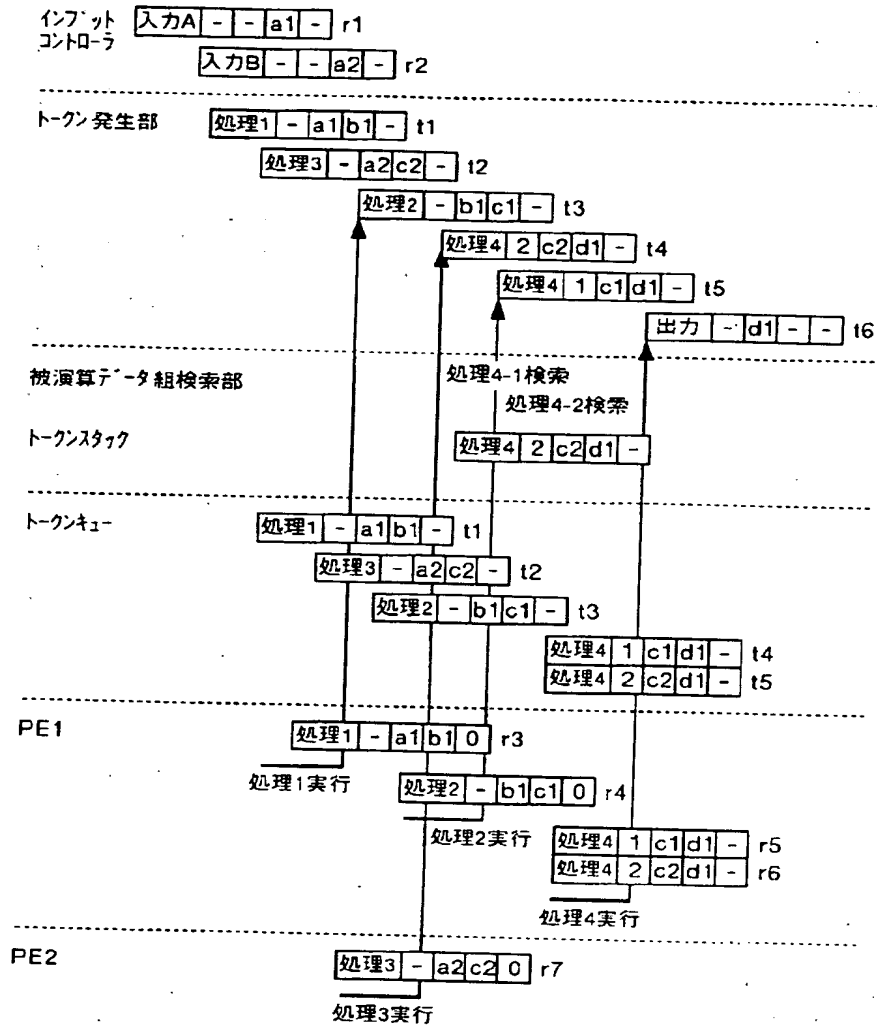
図 13





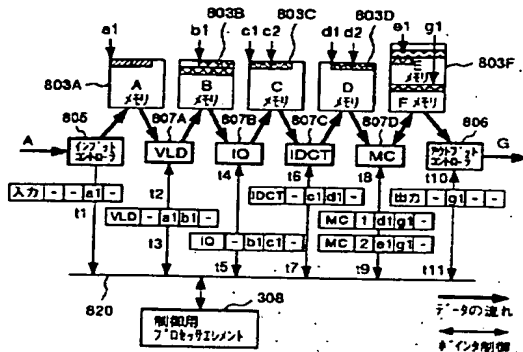
【図12】

図 12



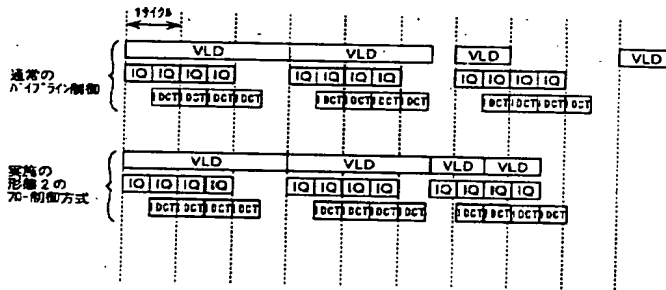
【図15】

図 15



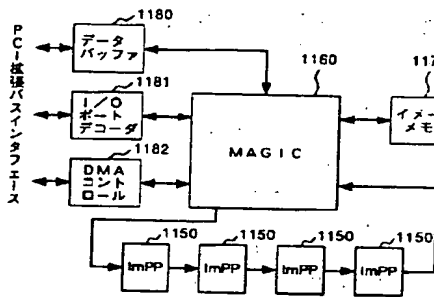
【図16】

図 16



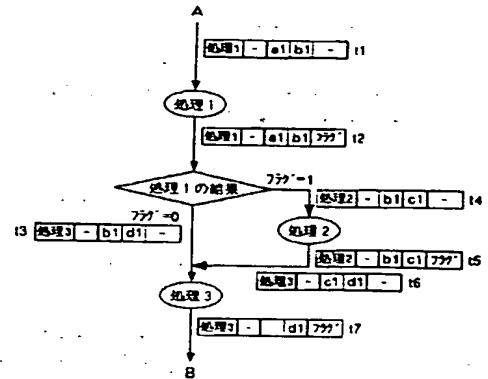
【図18】

図 18



【図17】

図 17



【図19】

図 19

